

DR-BART Implementation

```
variance = 'ux':  
  SCALE_MIX = TRUE  
variance = 'x':  
  SCALE_MIX = FALSE
```

```
std::vector<vec_d> xinfo;
```

xi : cutpoints mean

xiprec : cutpoints prec

Each element in the vector represents a variable.

Each variable has cutpoints

di : data info

contains the x data and also the y data

pinfo pi: Contains MCMC action probabilities

pbd : prob of birth / death

pd : prob of birth given birth / death

getpb : get probability of birth for a tree

returns 0 when there is no bottom node to split on

returns 1 when the tree is empty

else returns 0.5

Mean trees:

bdhet : birth-death heteroscedastic

- can change a mean tree by spawning / deleting new nodes
- First: Decide whether to birth or to death by using getpb
- if: birth operation; else: death operation
- For the birth operation:
 - Randomly (uniformly distributed) samples a possible node to split on (from goodbots)
 - Randomly (uniformly distributed) samples a possible variable to split on (from goodvars)
 - Randomly (-|-) samples a cutting point from the variable
 - Then calculates metropolis ratio: alpha
 - Therefore calls: getsuffhet
 - getsuffhet iterates over all samples to get [the numbers of samples for the two leaves (left and right) ???]
 - and the likelihood?
 - Therefore calls bn:
 - **Assumption: bn takes long when there are lots of u splits**
 - **But getsuffhet it seems to ignore the other variables it has not split on?**
 - If left or right samples < 5 : then do not birth, because the samples are too few
 - Then randomly samples probability (uniformly distributed): if it is smaller than alpha, then do the birth; else do not do it
- ...

drmuhet : draw mu heteroscedastic (model)

- changes a mean tree t
- sets leave means to means of samples considering the variances(?)

Precision trees:

bdprec:

drphi:

- changes a precision tree
- updates leave precisions
- uses the gig_norm function. If I recall correctly, this is an approximation of the normal distribution and was mentioned in the paper

[\[ⓘ\]](#) bdhet

- **Purpose:** Proposes and performs a **birth-death move** on the structure of the tree.
- **What it does:** It grows (birth) or prunes (death) nodes in the tree, changing the tree's topology (i.e., where and how it splits the data).
- **Why:** This allows the model to adaptively find the best tree structure for fitting the data, exploring different partitions of the predictor space.
- **Heteroscedastic context:** The "het" suffix means this operation is performed in a way that accounts for non-constant variance (heteroscedasticity) in the data.

[\[ⓘ\]](#) drmuhet

- **Purpose:** Draws or updates the **mean parameters** (μ) at the terminal nodes (leaves) of the tree.
- **What it does:** Given the current tree structure, it samples new values for the means at each leaf, typically from their posterior distribution, considering the current data and precision (variance) estimates.
- **Why:** This step updates the predicted values at the leaves, improving the fit of the model without changing the tree structure.
- **Heteroscedastic context:** Again, the "het" suffix means this update accounts for observation-specific variances.

Summary:

- [\[ⓘ\]](#) bdhet changes the **structure** of the tree (where it splits).
- [\[ⓘ\]](#) drmuhet updates the **values** at the leaves (the means), given the current structure.

Both are essential for fully exploring the posterior in a BART model with heteroscedasticity.



ltsstar

What does drmuhet stand for?

[drbarthetRcppClean.cpp:221](#)

GitHub Copilot

> Used 1 reference

[drmuhet](#)

stands for "**draw mu (mean) for heteroscedastic model**".

- **dr**: short for "draw", meaning to sample or update a parameter in a Bayesian MCMC context.
- **mu**: the mean parameter at the leaves (terminal nodes) of a regression tree.
- **het**: short for "heteroscedastic", indicating the model allows for non-constant variance (heteroscedasticity) across observations.

So, [drmuhet](#) is the function that samples (draws) new mean values for the leaves of a tree, taking into account heteroscedastic (observation-specific) variances.



Workspace

> Used 1 reference

The function `drmuhet` is responsible for updating the mean parameters (often called "mu") of the terminal nodes (leaves) in a regression tree, specifically in the context of a heteroscedastic Bayesian Additive Regression Trees (BART) model. In heteroscedastic models, the variance of the errors can change across observations, so this update must account for varying precision (inverse variance), which is passed in as the `phi` argument.

The function begins by declaring two containers: `bnv`, which will hold pointers to the bottom nodes (leaves) of the tree, and `sv`, a vector of sufficient statistics for each leaf. It then calls `allsuffhet`, a function that computes these sufficient statistics for the current tree structure, given the data and the heteroscedastic precisions.

Next, the function calculates a prior precision parameter `a` based on the prior standard deviation (`pi.tau`) and the prior variance `sig2` (though `sig2` is not used further in this function). For each leaf node, it computes the posterior variance (`fcvar`) and mean (`fcmean`) for the node's mean parameter, using the sufficient statistics. The update formula reflects Bayesian conjugacy for the normal-normal model, but with the node-specific sufficient statistics and the heteroscedastic precision incorporated.

For each leaf, the function samples a new mean from a normal distribution with the computed mean and variance, and assigns this value to the node using `setm`. As a safety check, it verifies that the new mean is not NaN (not a number); if a NaN is detected, it prints diagnostic information and stops execution with an error. This helps catch numerical issues during model fitting.

Overall, `drmuhet` is a key part of the MCMC update for BART models with non-constant variance, ensuring that each tree's leaf means are updated in a way that reflects both the observed data and the current estimates of heteroscedasticity.

Workspace

> Used 1 reference

The line

```
fit(t[j], xi, di, ftemp);
```

calls the `fit` function, which is a core operation in tree-based models like BART (Bayesian Additive Regression Trees). Here, `t[j]` refers to the `j`-th tree in an ensemble of regression trees. The arguments `xi` and `di` typically represent the predictor variables and associated data or metadata needed for prediction. The last argument, `ftemp`, is a vector that will be filled with the predicted values for each observation in the dataset, as computed by the current state of the tree `t[j]`.

In practice, this function traverses the tree `t[j]` for each observation, using the predictor variables in `xi` and the data information in `di` to determine which terminal node (leaf) each observation falls into. It then assigns the corresponding predicted value from that leaf to the appropriate position in `ftemp`. This process is repeated for all observations, so that after the function call, `ftemp` contains the predictions of tree `t[j]` for the entire dataset.

This operation is fundamental in ensemble methods, as it allows the model to aggregate predictions from multiple trees, update residuals, and perform other calculations necessary for model fitting and inference.



Revision #3

Created 4 June 2025 06:11:46 by Admin

Updated 4 June 2025 16:06:55 by Admin